PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : G06F 19/00 — A1

(11) International Publication Number: WO 99/59097

(43) International Publication Date: 18 November 1999 (18.11.99)

(21) International Application Number: PCT/AU99/00350

(22) International Filing Date: 11 May 1999 (11.05.99)

(30) Priority Data:
PP 3473 11 May 1998 (11.05.98) AU

(71) Applicant *(for all designated States except US)*: CREATIVE EDGE INTERNET SERVICES PTY. LTD. [AU/AU]; Suite 501, 56 Berry Street, North Sydney, NSW 2060 (AU).

(72) Inventor; and

(75) Inventor/Applicant *(for US only)*: WERKHOVEN, Richard, John [AU/AU]; 111 Wallis Street, Strathfield, NSW 2135 (AU).

(74) Agent: GRIFFITH HACK; Patent & Trade Mark Attorneys, G.P.O. Box 4164, Sydney, NSW 2001 (AU).

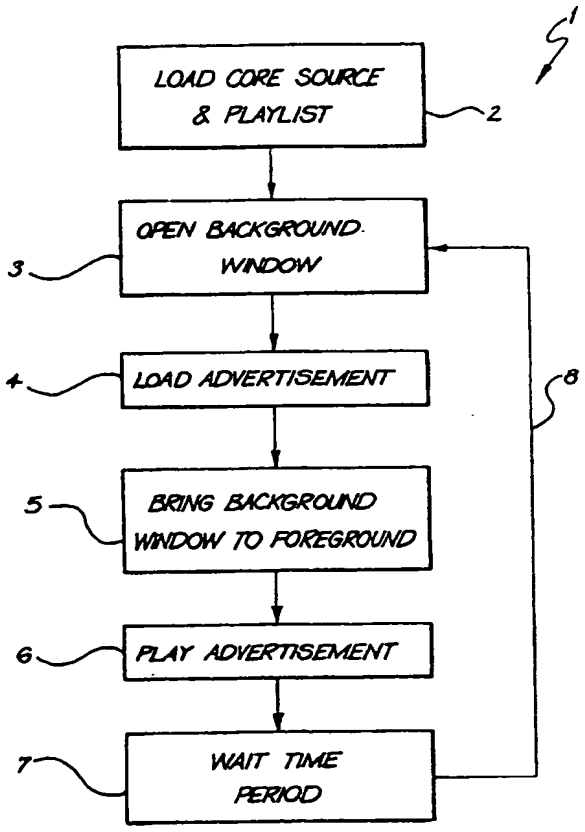
(81) Designated States: AE, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, US, UZ, VN, YU, ZA, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE). OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published
With international search report.

(54) Title: INTERNET ADVERTISING SYSTEM

(57) Abstract

In a computer user interface environment for the display of information, a method is disclosed of providing push content delivery comprising the steps of: (a) providing a popup window having determined content, the popup window being provided after a predetermined time of a user viewing predetermined information and the recording of the completion of content delivery where the user has not closed the abovementioned popup window prior to completion of the display of the determined content and (b) the window disappearing after a second predetermined interval.



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	**ES**	Spain	**LS**	Lesotho	**SI**	Slovenia
AM	Armenia	**FI**	Finland	**LT**	Lithuania	**SK**	Slovakia
AT	Austria	**FR**	France	**LU**	Luxembourg	**SN**	Senegal
AU	Australia	**GA**	Gabon	**LV**	Latvia	**SZ**	Swaziland
AZ	Azerbaijan	**GB**	United Kingdom	**MC**	Monaco	**TD**	Chad
BA	Bosnia and Herzegovina	**GE**	Georgia	**MD**	Republic of Moldova	**TG**	Togo
BB	Barbados	**GH**	Ghana	**MG**	Madagascar	**TJ**	Tajikistan
BE	Belgium	**GN**	Guinea	**MK**	The former Yugoslav Republic of Macedonia	**TM**	Turkmenistan
BF	Burkina Faso	**GR**	Greece			**TR**	Turkey
BG	Bulgaria	**HU**	Hungary	**ML**	Mali	**TT**	Trinidad and Tobago
BJ	Benin	**IE**	Ireland	**MN**	Mongolia	**UA**	Ukraine
BR	Brazil	**IL**	Israel	**MR**	Mauritania	**UG**	Uganda
BY	Belarus	**IS**	Iceland	**MW**	Malawi	**US**	United States of America
CA	Canada	**IT**	Italy	**MX**	Mexico	**UZ**	Uzbekistan
CF	Central African Republic	**JP**	Japan	**NE**	Niger	**VN**	Viet Nam
CG	Congo	**KE**	Kenya	**NL**	Netherlands	**YU**	Yugoslavia
CH	Switzerland	**KG**	Kyrgyzstan	**NO**	Norway	**ZW**	Zimbabwe
CI	Côte d'Ivoire	**KP**	Democratic People's Republic of Korea	**NZ**	New Zealand		
CM	Cameroon			**PL**	Poland		
CN	China	**KR**	Republic of Korea	**PT**	Portugal		
CU	Cuba	**KZ**	Kazakstan	**RO**	Romania		
CZ	Czech Republic	**LC**	Saint Lucia	**RU**	Russian Federation		
DE	Germany	**LI**	Liechtenstein	**SD**	Sudan		
DK	Denmark	**LK**	Sri Lanka	**SE**	Sweden		
EE	Estonia	**LR**	Liberia	**SG**	Singapore		

BNSDOCID: <WO___9959097A1_I_>

INTERNET ADVERTISING SYSTEM

Field of the Invention

The present invention relates to the utilization and delivery of content when utilizing a user interface on a computer and has application to advertising over the Internet as well as other forms of content delivery.

Background of the Invention

Recently, society has seen an explosion in the utilization of the Internet and other similar computer networks for the conveyancing information. In particular, the "World Wide Web", has provided for the cataloguing and accessing of almost an infinite amount of information.

While web pages were originally a vehicle limited to placing text on a screen on remote computers, web pages have effectively become a receptacle for sound, pictures, animations and a form of video, amongst other forms of information.

Further, recently, the World Wide Web has experienced a high degree of commercialization. It is now common to provide for advertising over the World Wide Web. Within any advertising program, one objective is to ensure the advertising is effective in placing the message before the viewer. Hence, the placement of appropriate advertising with certain Internet sites has grown up as a separate Internet industry with the resulting revenue from advertising often driving the production of web pages. Of course, with such developments as the convergence of the Internet with interactive television and the further convergence with computer operating systems, the utilization of advertising is becoming more important generally within such computer systems.

Despite innovations in Internet-related technology, there is often a significant delay between content being requested by a user from a provider and that requested information being displayed on the computer screen which can result in such requests being cancelled by users before the content can be displayed. One consequence of this for advertising is that many users fail to view the intended advertisements. The delay is often due to the bandwidth limitations of delivery. In practice, users are very sensitive to waiting for extended periods for content delivery.

Another shortcoming with existing form of Internet-based advertising is that, due to limitations of existing browsers and code in use, there is no way for the advertiser to determine if the user had closed the window containing the advertisement before the advertisement could complete its presentation.

Summary of the Invention

It is an object of the present invention to provide for improved content delivery capabilities with interactive computer systems and to enable the measurement of completion of that content being displayed on a user's computer screen.

In accordance with a first aspect of the present invention, there is provided in a computer user interface environment for the display of information, a method of providing push content to a user comprising the step of: (a) automatically displaying a pop-up window displaying the push content material, the pop-up window being provided a predetermined time after a user has begun viewing

BNSDOCID: <WO___9959097A1_I_>

first predetermined information.

The push content can be separately loaded over a network whilst the user can be viewing the first predetermined information. Preferably, the popup window disappears after a second predetermined interval. The method can further include the step of iterating step (a) after a third predetermined time interval.

The user interface can comprise an Internet browser and the information can be stored at an Internet site. Preferably, the method continues with the step (a) whilst a user visits pages within the Internet site.

The push content can be specific to the browser utilized by the user. The method can be implement throu: :tilization of a scripting language of the browser. The predetermined information can be varied in a. :nce with the time of access by the user.

In accordance with a further aspect of the present invention, there is provided in a computer user interface environment for the display of information, a method of providing push content delivery comprising the steps of: (a) providing a popup window having a determined content, the popup window being provided after a predetermined time a user viewing predetermined information, the pop up window further displaying second predetermined information; and (b) recording whether the popup window was closed by the user prior to completion of second interval and the display of the determined portion of content was completed. (c) closing or repositioning the popup window at the back of other windows after a third interval.

Preferably, the method further comprises iterating steps (a) to (c) after a fourth interval. The push content can further be varied in accordance with parameters available to the programming or scripting language used in a particular implementation of the method. The push content can be varied in accordance with a detected IP address of the user.

Brief Description of the Drawings

Notwithstanding any other forms which may fall within the scope of the present invention, preferred forms of the invention will now be described, by way of example only, with reference to the accompanying drawings in which:

Fig. 1 illustrates a basic flow chart for the steps of the preferred embodiment.

Description of Preferred and Other Embodiments

The preferred embodiment consists of a series of, or individual interactive web sites which deliver "popup" content to users visiting the World Wide Web page. These Web pages can be encoded utilizing standard HTML and Java Script encodings although it will be readily apparent to those skilled in the art that the present invention is readily applicable to many other language formats.

Preferably, this system delivers a predetermined portion of the content within a "popup" window which is provided for a predetermined time and then removed from the user's screen. The time is preferably set for each individual predetermined portion of the content and the software then waits for a predetermined period of time before running the next determined portion of content, even if a user switches pages within a site. If a user leaves the site completely, then the system is unable to run another

portion of content until the visitor returns to the site. In any event, preferably the system does not run the next determined portion of content until the necessary time has passed since the last determined portion of content. The preferred embodiment can be implemented utilizing a browser scripting language such as JavaScript and preferably waits before beginning to load the determined portion of content until after the main page has finished loading. This is more reliable than loading the page and determined portion of content simultaneously, and ensures that the site itself loads without interruption. This can, of course, be configured for each page/frameset running the software.

The software checks the version of the browser the visitor is using. If the browser supports it, the determined portion of content will be loaded into a window while it is in the background and then moved to the foreground, otherwise it will be loaded in the foreground.

The user is able to switch windows or close the window containing the determined portion of content, thereby skipping that particular determined portion of content - although the next determined portion of content in the sequence will still run. If the window is not closed by the user before a given determined portion of content has been completely displayed on the user's screen, then a record of that completion can be added to a tally recorded in a predetermined file.

The runtime of each determined portion of content is determined by the determined portion of content itself - this allows for more flexibility in the design of the content to be delivered and the ability for the determined portions of content to change their length depending on circumstances.

The sequence of determined portions of content and the timing of the gaps between them can be determined by a playlist.

The playlist for the determined portions of content along with the code to run them must currently be included in each page that the determined portions of content are to run from. This either has to be auto-inserted by the server or added to the content of each page.

The only exception to this is framed sites, where the code & playlist can be run from the page declaring the frameset and will then apply to all pages in the frameset.

As an alternative, it would be possible for the code and playlist to reside in separate files that are referenced from the pages requiring them, but this part of JavaScript is not supported by some versions of Internet Explorer currently in use. This situation will change as users move to newer versions of browsers.

This method of insertion is likely to reduce site management overhead as well as reduce the effective size of the code for each page, and the JavaScript and playlist is likely to be cached separately by the user's browser as well as by the proxy service they are using.

This system is preferable to any system which opens an empty window every time a user attempts to view the top level (home page) of a site before loading the determined portion of content and remains on screen until closed by the visitor - this either results in the user closing the window before the determined portion of content has finished loading, or multiple windows are left on the screen all showing the same determined portion of content.

Turning now to Fig. 1, there is illustrated a basic example flow chart of this steps 1 of

the preferred embodiment. Initially, when a user opens a Web Page at a site, the poor information for that Web Page is downloaded 2 in addition to a playlist of popup advertisements.

Next, HTML code is instructed to open a background window and the advertisement is loaded from its relevant HTML source 4. Upon loading, the add is brought to the foreground 5 and "played" 6. Subsequently, a time period lapses 7 and the method of the preferred embodiment iterates 8 back to the step 3.

Whilst an actual example of the relevant HTML encoding is provided in the attached appendix A, a number of general parts of this code will now be described.

To start the sequencer the following is added to the html <BODY> tag.

<BODY onLoad - "startNetBreak()">

Playlist

The playlist can be in the following format

"Playlist Start

Array Declarations

itemURL [0] = "*URL of first item*"

itemWait [0] = *seconds before first item*

itemSize [0] = "*width=width in pixels of first item, height = height in pixels of first ad*"

itemURL [1] = "*URL of item 2*"

itemWait [1] = *seconds before item 2*

itemSize [1] = "*width=width in pixels ad 2. height=height in pixels of item 2*"

.....

itemURL [n-1] = "*URL of item n*"

itemWait[n-1] = *seconds before item n*

itemSize [n-1] = "*width=width in pixels item n. height=height in pixels and item n*"

' Playlist End

Here is a sample playlist for determined portions of content.

'Playlist Start

var itemURL = new Array (2)

var itemWait = new Array (2)

var itemSize = new Array (2)

itemURL [0] = http://netbreak.com.au/Popups/EdgeLogoSeq.html"

itemWait [0] =60

itemSize [0] = "width=620, height=420"

itemURL [1] = "http:// -netbreak.com.au/Popups/PromoTester.html"

itemWait [1] = 60

itemSize [1] = "width=200, height=150"

' Playlist End

BNSDOCID: <WO___9959097A1_I_>

There are a number of different methods of implementing this system on a web site. The implementation can be dependent on the way the web site is being served and the capabilities of the web server in use.

1. Live database generated web pages

The database system generating the pages would insert the JavaScript and Playlist into the required pages as the pages are generated.

This would only require modification one file when the playlist is changed and the page content would then be updated for all new pages generated.

2. Scriptable web server

The web server could automatically insert the JavaScript and Playlist into the required pages as it is serving the pages.

This option would also require only one change when the playlist is changed.

3. Straight web serving - no server programming

The JavaScript and Playlist block can be inserted into the pages by editing the HTML file for each page.

This would require each page to be edited when the playlist is changed.

The system is preferably capable of running any content that can be handled by the browser, as it can display the determined content by loading a URL into the popup window. The content can be responsible for bringing itself to the front when loaded.

The window is closed when the content signals to the originating window that it has finished. Therefore for the window to go away automatically requires the insertion of a small JavaScript to send this message and also requires a call to tell this script when to do so.

As a result of this, although any URL can be used it may be necessary to add JavaScript to each determined portion of content so that it presents correctly.

A variety of further refinements can be implemented in certain configurations. These include firstly that the JavaScript code, when used, can be created to selectively load contents based on the capabilities of the user's browser and plug-ins, enabling the use of plug-in dependent content where possible and at the same time ensuring content delivery by delivering an alternate version where necessary. An example of such a Browser capability change is given in the Appendix Example.

In a second refinement the selective content ability can also be used to target content specifically for the user, as long as the necessary information is available to the browser. This feature can tie in with information based on what pages the user has visited or on forms data collected by adding JavaScript to the pages collecting the data. This could also be used to advertise browsers or plug-ins for example, depending on what the user already has –informing the user of an update, for example.

In a third refinement, the time interval for the display of the predetermined portion of content can be determined by rules encoded into the content being displayed. These rules can be dependent upon such parameters as mouse clicks, keyboard events, the type of browser user by the user, the hardware used by the user and any other parameters available to the programming or scripting

language used in a particular implementation of this system.

In a fourth refinement, if another window (or windows) is (or are) brought in front of the popup window displaying the determined portion of content, the popup window will automatically return to the frontmost position after a predetermined portion of time. This can be implemented as part of playing an advertisement or as part of the playlist loop.

In a fifth refinement, where a further portion of content is to be delivered for display in a popup window that has completed the display of a determined portion of content, the popup window will automatically return to the rearmost position until the new portion of content is ready to be displayed in the popup window, after which the popup window will automatically return to the frontmost position and display the new portion of content.

In a sixth refinement, the popup window can be made to 'popup' on screen in the frontmost position at predetermined times of day and/or on predetermined dates.

In a seventh refinement, the predetermined portion of content can be determined by rules encoded into software residing on the file server management hardware providing the site implementing the system described in this document. These rules can be dependent upon such parameters as the type of browser user by the user, the hardware used by the user, the IP address of the device requesting the file.

In an eighth refinement, software residing on the file server management hardware providing the site implementing the system described in this document can determine the content according to the bandwidth available to the user, derived from information in the IP address or domain of the device requesting the file. This can be used to deliver larger file sizes or different media types to high-bandwidth connections.

In a ninth refinement, software residing on the file server management hardware providing the site implementing the system described in this document can determine the content according to the location of the user, derived from information in the IP address or domain of the device requesting the file. This can be used to deliver localized information such as local weather or specific-language information, for example.

In a tenth refinement, software residing on the file server management hardware providing the site implementing the system described in this document can determine the content according to the user's domain-specific information, derived from information in the IP address or domain of the device requesting the file. This can be used to deliver domain-specific information such as educational information to educational sites (.edu) which uses information from the top level of the domain information, or advertising targeting users of a particular Internet service provider (.domain.com), which would use secondary as well as top level domain information, for example.

In an eleventh refinement, the popup window and the predetermined content can be subject to combinations of the abovementioned refinements.

Ideally the content used in the popup window should be kept to as few files as possible and should be able to load in about 30 seconds. At present standard modem bandwidth limits, this would

probably mean a file size of about 150Kb with modern computer modems (56K) at most unless there is a good chance that a user will be on a page for more than long enough for the page to load.

Ideally the content to be displayed in the popup window should be small enough to fit a 640x480 pixel screen with menu bar, window frame, title bar and the extra space that the browser leaves from the left edge of the window. It is therefore suggested that the maximum size is 600(horizontal)x400(vertical) pixels to ensure good screen fit.

It is also desirable to use a standard size for all portions of content across a site - if not across all sites to ensure visitor comfort and reduce time taken for visitors to adjust to the appearance of the window. 540(horizontal)x405(vertical) pixels would provide a sufficient screen area for the advertisement while sitting comfortably within a 640(horizontal)x480(vertical) screen.

Of course, many modifications are possible. For example, the type of content used may be varied in accordance with the current time zone of the user. For example, different night time and day time content might be provided.

It would be appreciated by a person skilled in Internet-related technologies that numerous variations and/or modifications may be made to the present invention as shown in the specific embodiments without departing from the spirit or scope of the invention as broadly described. The present embodiments are, therefore, to be considered in all respects to be illustrative and not restrictive.

Appendix A - Example HTML code

```
<HTML>
<HEAD>
</HEAD>
<BODY onLoad="startNB()">

<H1>Code for Anzwers Ads</H1>

<H1>Release version.</H1>
<P>
Playlist is:<P>
<P>
Anzwers01 - 20 Hours Delay<P>
Anzwers02 - 40 Hours Delay<P>
Anzwers01 - 20 Hours Delay<P>
Anzwers02 - 60 Hours Delay<P>
Anzwers01 - 60 Hours Delay  <---- Loop back to here<P>
Anzwers02 - 60 Hours Delay<P>
<P>


<!-- begin NetBreak -->
<!-- ASX Release 1 -->
<SCRIPT LANGUAGE = "JavaScript">
<!-- begin script
var alertID = null;  // NetBreak(TM) System V1.1 (C)Creative Edge Internet Services
var curNB = 0;  // Patent Pending in various territories
var NBCook;
var delCook;
var delCK = "_Delay";
var seqCK = "_Next";
var NBplName;
var NBDelay;
var NBckExp;
var expire;
var curTime;
var expDelay;
var NBDelay = 0;
```


BNSDOCID: <WO___9959097A1_I_>

```
var NBLoopTo;
var ckDomain;
var NBPage = new Array(1); // Keywords for URLs of pages
var NBURL = new Array(1); // URLs of NBs
var NBWait = new Array(1); // Time before
var NBSize = new Array(1); // Window size
function getCookieVal (offset) {
 var endstr = document.cookie.indexOf (";", offset);
  if (endstr == -1)
   endstr = document.cookie.length;
 return unescape(document.cookie.substring(offset, endstr+1));
}
function FixCookieDate (date) {
 if(navigator.appVersion.indexOf("2.") != -1) {
  var base = new Date(0);
  var skew = base.getTime();
  if (skew != 0)
   date.setTime (date.getTime() - skew);
      }
}
function GetDateStr (date) {
 var dateS = date.toString();
 if(dateS.indexOf("(") != -1) {
  dateS = dateS.substring(0, dateS.indexOf("(")) +  dateS.substring(dateS.indexOf(")") + 1,
dateS.length);
 }
return dateS;
}
function GetCookie (name) {
 var arg = name + "=";
 var alen = arg.length;
 var clen = document.cookie.length;
 var i = 0;
 while (i < clen) {
  var j = i + alen;
  if (document.cookie.substring(i, j) == arg)
  return getCookieVal (j);
  i = document.cookie.indexOf(" ", i) + 1;
```


BNSDOCID: <WO___9959097A1_I_>

```
 if (i == 0)
  break;
 }
 return null;
}
function SetCookie (name,value,expires,path,domain,secure) {
 if(expires) {
 expires.setTime(expires.getTime()  + (3600000));
 }
 document.cookie = name + "=" + escape (value) +
  ((expires) ? "; expires=" + expires.toGMTString() : "") +
  ((path) ? "; path=" + path : "") +
  ((domain) ? "; domain=" + domain : "") +
  ((secure) ? "; secure" : "");
}
function NBCheckURL () {
 var i = 0;
 if(navigator.appVersion.indexOf("2.") == -1) {
 while (i < NBPage.length) {
  if (location.href.toLowerCase().indexOf(NBPage[i].toLowerCase()) != -1) {
   return 1;
   break;
  }
  i = i + 1;
 }
 }
 return null;
}
function startNB() {
 if(NBCheckURL()) {
 delCK = NBplName+"_Delay";
 seqCK = NBplName+"_Next";
 expire = new Date();
 curTime = new Date();
 expDelay = expire.getTime()  + (NBckExp);
 expire.setTime(expDelay);
  NBCook = GetCookie (seqCK);
  if(NBCook) {
```


BNSDOCID: <WO___9959097A1_I_>

```
 curNB = parseInt(NBCook);
 }
 if(curNB >= NBURL.length) {
 curNB = NBLoopTo;
 SetCookie (seqCK,curNB,expire,"/",ckDomain);
 }
 delCook = GetCookie (delCK);
 if(delCook) {
 curTime = new Date();
 NBDelay = Date.parse(delCook) - curTime.getTime();
 }
 if((NBDelay) <= 200)
    NBDelay = 200;
 if (NBDelay < 100000)
 alertID=setTimeout("displayNB()", NBDelay);
 }
}
function delayNB() {
 NBDelay = NBWait[curNB]*1000;
 var nextTime = new Date();
 var NBTime = nextTime.getTime()  + (NBDelay);
 nextTime.setTime(NBTime);
 SetCookie (delCK,GetDateStr(nextTime),nextTime,"/",ckDomain);
 if (NBDelay < 100000)
 alertID=setTimeout("displayNB()", NBDelay);
}
function displayNB() {
 SetCookie (seqCK,curNB+1,expire,"/",ckDomain);
 delayNB();
 NBWin=window.open(NBURL[curNB]+"?"+"h="+location.hostname+"+p="+location.pathname,"
 NB"+curNB,NBSize[curNB]+",toolbar=0,location=0,directories=0,status=0,menubar=0,scrollbars
 =0,resizable=0");
 if(NBWin == null) {
 NBWin=window.open(NBURL[curNB]+"?"+"h="+location.hostname+"+p="+location.pathname,"
 NB"+curNB,NBSize[curNB]+",toolbar=0,location=0,directories=0,status=0,menubar=0,scrollbars
 =0,resizable=0");
 }
 if(parseInt(navigator.appVersion) > 3) {
```


BNSDOCID: <WO___9959097A1_I_>

```
 focus();
}
curNB += 1;
if(curNB >= NBURL.length)
 curNB = NBLoopTo;
}
<!-- begin Config -->
NBckExp = 2678400000;  // sequence cookie expire time
NBLoopTo = 4;  // Point in playlist to loop back to
curNB = 0;  // First NB to run if no cookie
ckDomain = null; // Domain for timing & sequencing cookies
<!-- end Config -->
<!-- begin PageKey -->
NBPage[0] = ""
<!-- end PageKey -->
<!-- begin PlayList -->
NBplName = "ASX01"; // cookie name for playlist
NBURL[0] = "http://nb1.netbreak.com.au/ASX/Anzwers01.html";
NBWait[0] = 72000; //  20 Hours
NBSize[0] = "width=245,height=170";
NBURL[1] = "http://nb1.netbreak.com.au/ASX/Anzwers02.html";
NBWait[1] = 144000; //  40 Hours
NBSize[1] = "width=245,height=170";
NBURL[2] = "http://nb1.netbreak.com.au/ASX/Anzwers01.html";
NBWait[2] = 72000; //  20 Hours
NBSize[2] = "width=245,height=170";
NBURL[3] = "http://nb1.netbreak.com.au/ASX/Anzwers02.html";
NBWait[3] = 216000; //  60 Hours
NBSize[3] = "width=245,height=170";
NBURL[4] = "http://nb1.netbreak.com.au/ASX/Anzwers01.html";
NBWait[4] = 216000; //  60 Hours
NBSize[4] = "width=245,height=170";
NBURL[5] = "http://nb1.netbreak.com.au/ASX/Anzwers02.html";
NBWait[5] = 216000; //  60 Hours
NBSize[5] = "width=245,height=170";
<!-- end PlayList -->
// end script -->
</SCRIPT>
```


BNSDOCID: <WO___9959097A1_I_>

```
<!-- end NetBreak -->


</BODY>
</HTML>
----------

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2//EN">
<HTML>
<HEAD>
<META HTTP-EQUIV="expires" CONTENT="1">
<META HTTP-EQUIV="Pragma" CONTENT="no-cache">
<META HTTP-EQUIV="refresh" CONTENT="240;URL=Anzwers01t.html">
<TITLE>Loading...</TITLE>
</HEAD>
<BODY BGCOLOR="#FFFFFF" onLoad="NBEnd()">
<CENTER>
<SCRIPT LANGUAGE="JavaScript">

<!-- hiding
var fType = "gif";  // NetBreak(TM) (C) Creative Edge Internet Services - Patent Pending
var bType = "d";
var running = 0;
var alertID = null;

if (navigator.userAgent && navigator.userAgent.indexOf("MSIE")>=0) {
 if(parseInt(navigator.appVersion) > 3) {
  blur();
bType = "";
 }
} else if (parseInt(navigator.appVersion) > 2) {
  blur();
bType = "";
}
var NBNext = "Anzwers01"+fType+bType+".html";
var ShockMode = 0;  // Using Portions of AfterShock © Macromedia
if (navigator.mimeTypes && navigator.plugins["Shockwave Flash"] &&
navigator.mimeTypes["application/x-shockwave-flash"].enabledPlugin) {
```


BNSDOCID: <WO___9959097A1_I_>

```
 fType = "swf";
} else if (navigator.userAgent && navigator.userAgent.indexOf("MSIE")>=0) {
 if ((navigator.userAgent.indexOf("Windows 98")>=0 || navigator.userAgent.indexOf("Windows
95")>=0 || navigator.userAgent.indexOf("Windows NT")>=0)) {
 document.write('<SCRIPT LANGUAGE=VBScript\> \n');
 document.write('on error resume next \n');
 document.write('ShockMode =
(IsObject(CreateObject("ShockwaveFlash.ShockwaveFlash.3")))\n');
 document.write('</SCRI'+'PT\> \n');
 }
 if ( ShockMode ) {
 fType = "swf";
 }
}

NBNext = "Anzwers01"+fType+bType+".html";
if(bType == "d")  {
 NBEnd();
}
if(fType == "gif") {
 document.write('<IMG SRC="http://www.zipworld.com.au/~cedi/popups/Anzwers01d.gif"
WIDTH=230 HEIGHT=150 ALT="Loading..." Border=0>');
}if(fType == "swf") {
 document.write('<EMBED SRC="http://www.zipworld.com.au/~cedi/popups/Anzwers01.swf"
WIDTH=230 HEIGHT=150 PLAY="false" LOOP="false" QUALITY="high"
SWLIVECONNECT="false"></EMBED>');
}
function NBEnd() {
 window.location.href =  NBNext+window.location.search;
}
function NBClick() {
 running = 0;
 NBNext = "Anzwers01dr.html";
 NBEnd();
}


// STOP -->
```


BNSDOCID: <WO___9959097A1_I_>

```
</SCRIPT>
</CENTER>
</BODY>
</HTML>
```

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2//EN">
<HTML>
<HEAD>
<META HTTP-EQUIV="expires" CONTENT="1">
<META HTTP-EQUIV="Pragma" CONTENT="no-cache">
<TITLE>Anzwers</TITLE>
</HEAD>
<BODY BGCOLOR="#FFFFFF" onLoad="doRun()">
<CENTER>
<EMBED SRC="http://www.zipworld.com.au/~cedi/popups/Anzwers01.swf" WIDTH=230
HEIGHT=150 PLAY="true" LOOP="false" QUALITY="high"
SWLIVECONNECT="false"></EMBED>
</CENTER>
</BODY>
<SCRIPT LANGUAGE="JavaScript">

<!-- hiding
var alertID = null;  // NetBreak(TM) (C) Creative Edge Internet Services - Patent Pending
var delayID = null;
var running = 1;
var runCK = "NB_Running";
var ckDomain = null;



var NBNext = 'Anzwers01swfe.html';
function doRun() {
 keepFront();
 alertID=setTimeout("NBEnd()",  35 * 1000);
```


BNSDOCID: <WO___9959097A1_I_>

```
}
function keepFront() {
 if(running == 1) {
  focus();
 }
 if(1 > 0) {
 delayID=setTimeout("keepFront()", 1 * 1000);
 }
}

function NBEnd() {
 running = 0;
 blur();
 // SetCookie (runCK,"",null,"/",ckDomain);
 window.location.href =  NBNext+window.location.search;
}
function NBClick() {
 running = 0;
 NBNext = "Anzwers01swfr.html";
 NBEnd();
}


// STOP -->

</SCRIPT>
</HTML>

-----------------------------------------

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2//EN">
<HTML>
<HEAD>
<META HTTP-EQUIV="expires" CONTENT="1">
<META HTTP-EQUIV="Pragma" CONTENT="no-cache">
<META HTTP-EQUIV="refresh" CONTENT="10;URL=Anzwers01swff.html">
<TITLE>Anzwers</TITLE>
<BODY BGCOLOR="#FFFFFF" onLoad="doRun()">
```


BNSDOCID: <WO___9959097A1_I_>

```
 
</BODY>
<SCRIPT LANGUAGE="JavaScript">

<!-- hiding
function doRun() {
 window.close();
// NetBreak(TM) (C) Creative Edge Internet Services - Patent Pending
}
// STOP -->

</SCRIPT>
</HTML>

----------

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2//EN">
<HTML>
<HEAD>
<META HTTP-EQUIV="expires" CONTENT="1">
<META HTTP-EQUIV="Pragma" CONTENT="no-cache">
<META HTTP-EQUIV="refresh" CONTENT="2;URL=Anzwers01swff.html">
<TITLE>Anzwers</TITLE>
</HEAD>
<BODY BGCOLOR="#FFFFFF" onLoad="doRun()">
<P>Loading...
</BODY>
<SCRIPT LANGUAGE="JavaScript">

<!-- hiding
function doRun() {
var
adWin=window.open("http://www.anzwers.com.au/","NetBreakReferer","width=620,height=370,t
oolbar=1,location=1,status=1,menubar=1,scrollbars=1,resizable=1");
if(adWin == null) {

adWin=window.open("http://www.anzwers.com.au/","NetBreakReferer","width=620,height=370,t
oolbar=1,location=1,status=1,menubar=1,scrollbars=1,resizable=1");
```


BNSDOCID: <WO___9959097A1_I_>

```
}
 window.location.href = 'Anzwers01swfe.html';
}
// STOP -->

</SCRIPT>
</HTML>
```

BNSDOCID: <WO___9959097A1_I_>

We Claim:

1. In a computer user interface environment for the display of information, a method of providing push content to a user comprising the step of:

(a) automatically displaying a pop-up window displaying the push content material, said pop-up window being provided a predetermined time after a user has begun viewing first predetermined information.

2. A method as claimed in claim 1 wherein said push content is separately loaded over a network whilst said user is viewing said first predetermined information.

3. A method as claimed in claim 1 wherein said window disappears after a second predetermined interval.

4. A method as claimed in claim 1 further comprising the step of iterating step (a) after a third predetermined time interval.

5. A method as claimed in any previous claim wherein said user interface comprises an Internet browser and said information is stored at an Internet site.

6. A method as claimed in claim 5 wherein said method continues with said step (a) whilst a user visits pages within said Internet site.

7. A method as claimed in any previous claim wherein said push content is specific to the browser utilized by said user.

8. A method as claimed in any previous claim wherein said method is implement through the utilization of a scripting language of said browser.

9. A method as claimed in any previous claim wherein said predetermined information is varied in accordance with the time of access by said user.

10. In a computer user interface environment for the display of information, a method of providing push content delivery comprising the steps of:

(a) providing a popup window having a determined content, said popup window being provided after a predetermined time a user viewing predetermined information, said pop up window further displaying second predetermined information: and

(b) recording whether the popup window was closed by the user prior to completion of second interval and the display of the determined portion of content was completed.

(c) closing or repositioning said popup window at the back of other windows after a third interval.

12. A method as claimed in claim 10 further comprising the step:

(d) iterating steps (a) to (c) after a fourth interval.

13. A method as claimed in claim 12 wherein said method reiterates said steps (a) and (c) whilst a user visits pages within said Internet site.

14. A method as claimed in any previous claim wherein said push content is varied in accordance with the time of access by said user.

15. A method as claimed in any previous claim wherein said push content is

BNSDOCID: <WO___9959097A1_I_>

varied in accordance with parameters available to the programming or scripting language used in a particular implementation of said method.

16. A method as claimed in any previous claim wherein said push content is varied in accordance with a detected IP address of said user.

BNSDOCID: <WO___9959097A1_I_>

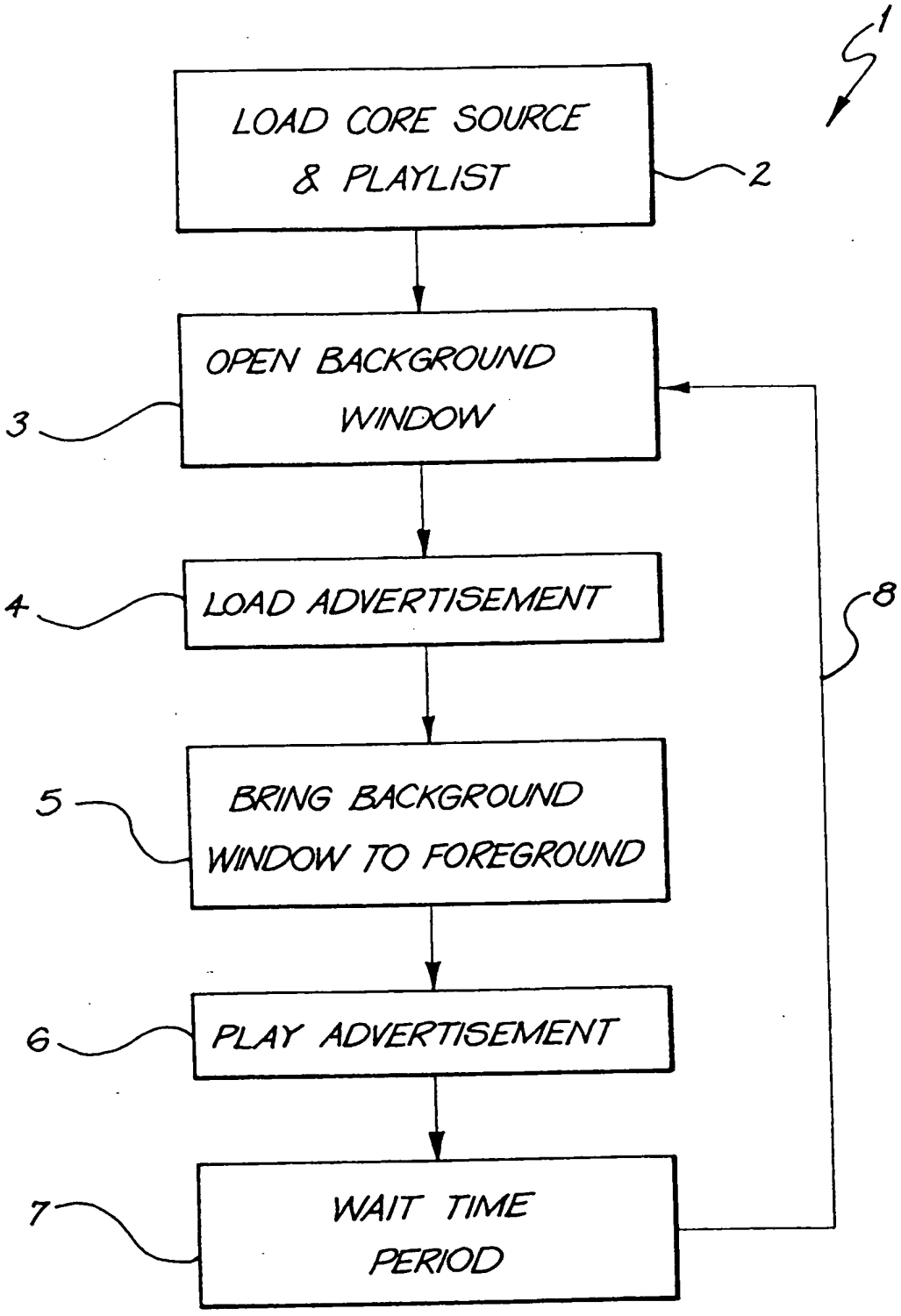


FIG. 1

BNSDOCID: <WO___9959097A1_I_>

INTERNATIONAL SEARCH REPORT

International application No.
PCT/AU 99/00350

A. CLASSIFICATION OF SUBJECT MATTER

Int Cl[6]: G06F 19/00

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
Int Cl.G06F 19/00

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
AU: IPC as above

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
USPTO: ICL/G06F$ AND ADVERTIS$
SOFTWARE PATENT INSTITUTE DATABASE: (Internet OR Web) AND (advertise)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
P,X	US-A-5890172 (Borman et al) 30 March 1999 See column 10 line 51 to column 11 line 16	1
X	WO-A-97/40601 (JUNO ONLINE SERVICES, L.P.) 30 October 1997 See page 10 lines 8 to 22	1, 4, 7
X	WO-A-97/40514 (JUNO ONLINE SERVICES, L.P.) 30 October 1997 See page 26 lines 3 to 12	1, 4, 7

[X] Further documents are listed in the continuation of Box C

[X] See patent family annex

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
11 June 1999	22 JUN 1999
Name and mailing address of the ISA/AU AUSTRALIAN PATENT OFFICE PO BOX 200 WODEN ACT 2606 AUSTRALIA Facsimile No.: (02) 6285 3929	Authorized officer **Michael Hardy** Telephone No.: (02) 6283 2547

BNSDOCID: <WO___9959097A1_I_>

INTERNATIONAL SEARCH REPORT

International application No.
PCT/AU 99/00350

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	WO-A-97/40447 (JUNO ONLINE SERVICES, L.P.) 30 October 1997 See page 11 line 3 to page 14 line 26	1, 4, 7
Y	Special Edition Using HTML 3.2, Third Edition, Mark R. Brown & Jerry Honeycutt, 1997 Que Corporation, pages 493 to 516, "ActiveX Controls" See page 504, "Timer Control"	1
Y	US-A-5572643 (Judson) 5 November 1996 See column 2 lines 6 to 11	1

BNSDOCID: <WO___9959097A1_I_>

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/AU 99/00350

This Annex lists the known "A" publication level patent family members relating to the patent documents cited in the above-mentioned international search report. The Australian Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

Patent Document Cited in Search Report		Patent Family Member					
US	5890172						
WO	97/40601	AU	24562/97	US	5838790		
WO	97/40514	AU	26113/97	EP	894329	US	5809242
WO	97/40447	AU	24552/97	EP	900422	US	5848397
US	5572643	AU	74588/96	AU	699439	CA	2235014
		EP	856180	WO	97/15020	US	5737619

END OF ANNEX

BNSDOCID: <WO___9959097A1_I_>

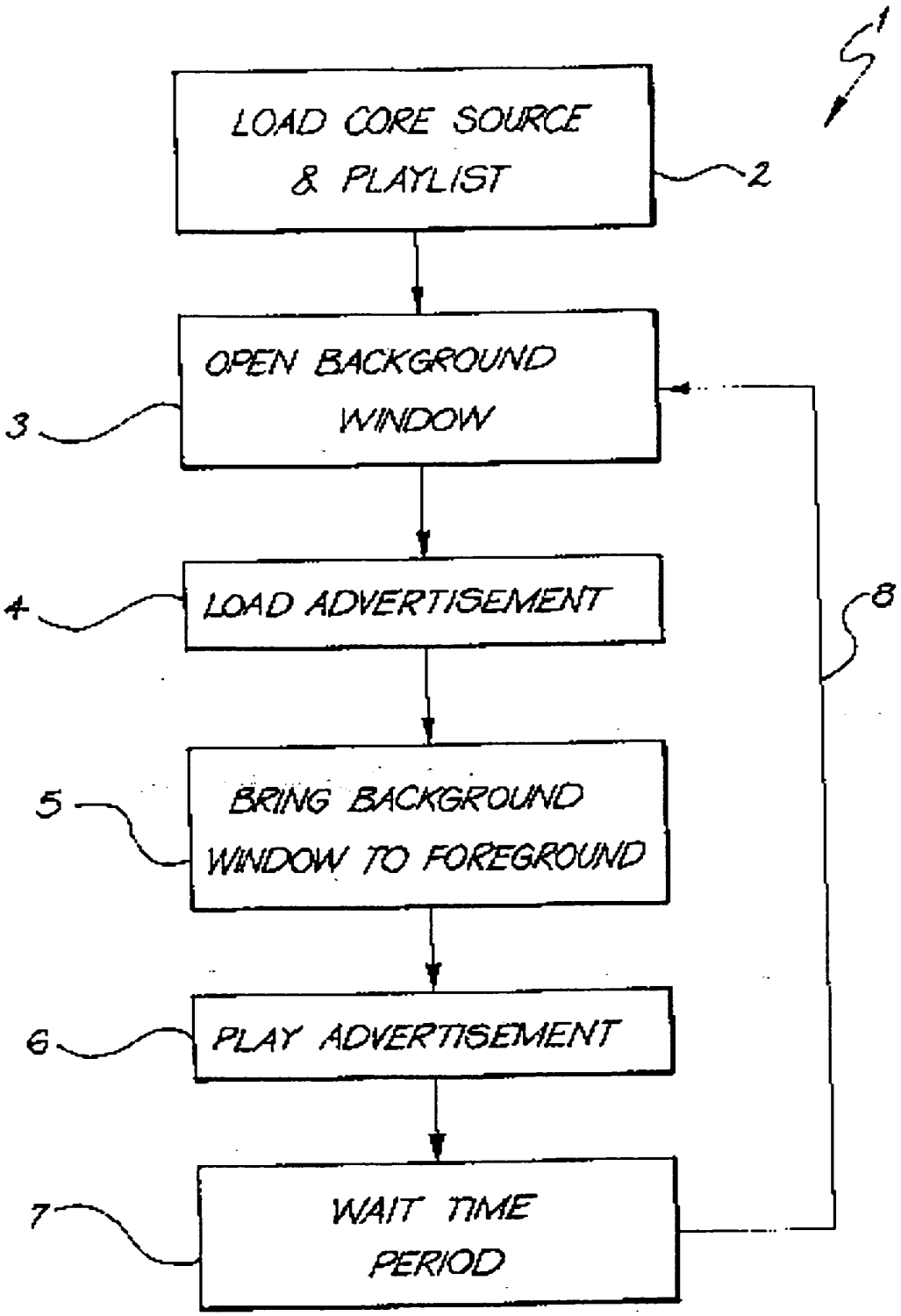


FIG. 1

BNSDOCID: <WO___9959097A1TI_>

THIS PAGE BLANK (USPTO)